

## COOPERATIVE ANALYSIS EXPERT SITUATION ASSESSMENT RESEARCH

Michael G. McCown, 1Lt, USAF  
Rome Air Development Center

For the past few decades, Rome Air Development Center has been conducting research in the area termed Artificial Intelligence. When the recent advances in hardware technology made many Artificial Intelligence techniques practical, the Intelligence and Reconnaissance Directorate of RADC initiated an applications program entitled Knowledge Based Intelligence Systems (KBIS). The goal of the program is the development of a generic Intelligent Analyst System, an open machine with the framework for intelligence analysis, natural language processing, and man-machine interface techniques, only needing the specific problem domain knowledge to be operationally useful.<sup>1</sup>

On the path towards such an architecture, RADC first implemented a number of domain specific expert systems. The evolutionary design of these individual expert systems has rapidly become more modular and distributed in itself (Figure 1). The next step of the generic design requires that the modular and distributed natures of these systems be carried to their necessary conclusion, that of loosely coupled, distributed, cooperating knowledge based system architecture. The effort to explore this step is entitled Cooperative Analysis Expert Situation Assessment Research (CAESAR).

In order to understand the evolution of the current architecture used in the KBIS program, it is necessary to understand a few fundamentals of intelligence analysis and the intelligence community. Much day to day intelligence analysis is done by assessing the status of indicators. Indicators are sets of actions that an enemy would be expected to take in preparation for an aggressive act. An indicator list is a list of activities that an enemy might be expected to engage in if they intended to initiate hostilities. The activities are logical/plausible moves or acts based on: a) Operational procedures, b) Observed activities during past conflicts and crises, and c) Results of intelligence assessments of enemy strategic offensive military doctrine.

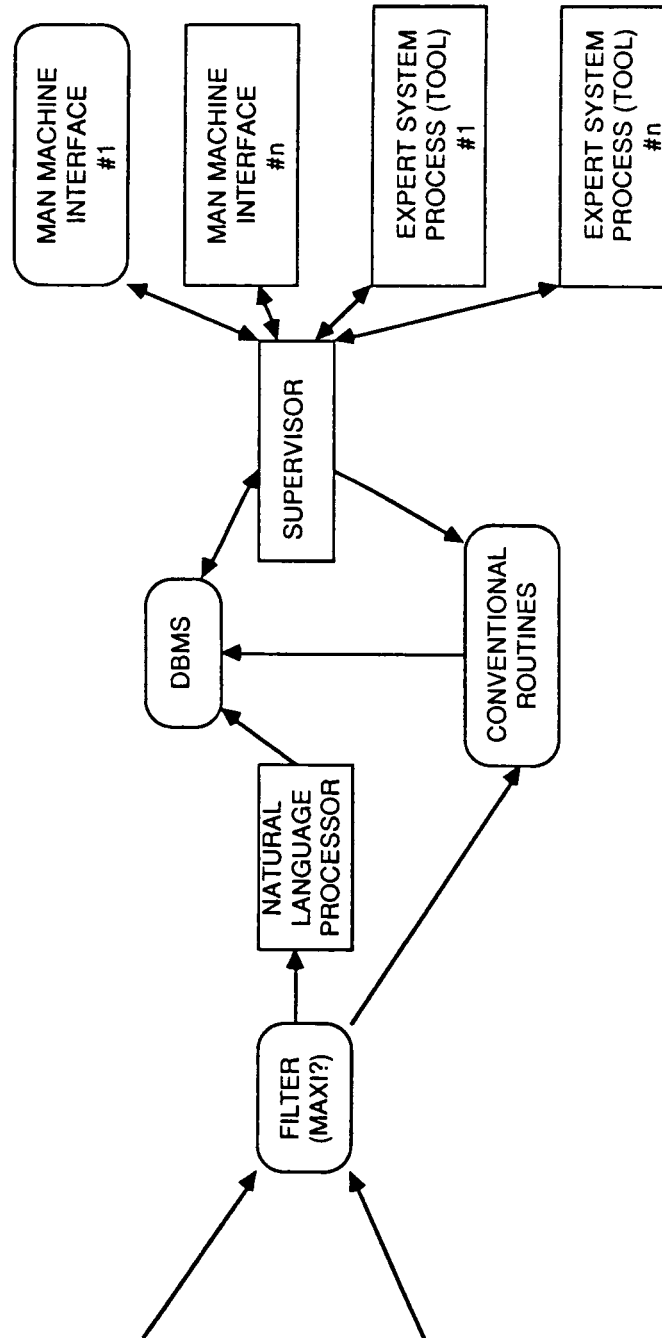
The monitoring of indicators pertaining to strategic warning is an extremely time-critical process since this warning impacts the degree of readiness of our

strategic offensive and defensive forces. Given the increasing threat from a number of sources, the task of providing an accurate and timely assessment of enemy intent is limited by the availability of qualified intelligence analysts and their ability to process increasingly large amounts of information in a crisis. While some tools exist to assist the intelligence analysis function (the World Wide Indicator Management System, for example), further research is needed to increase the accuracy and timeliness of strategic warning.

Today's automated support for the intelligence analyst is limited primarily to Intelligence Data Handling Systems, which often times can be as difficult to use as the paper pushing methods they replace. Yet automated data handling support is dictated by the massive volumes of traffic being passed through all Indications and Warning Centers. In addition to the data flow and management problem, the analysis and fusion of that data is becoming increasingly complex and subtle as foreign technology and doctrine improves. Today's intelligence analyst requires phenomenal memory, computer skills, and analytical capability in order to be effective. To relieve some of the burden, automated support must take the role of an informed colleague to the analyst, extending his memory, performing data retrievals, and aiding him in analysis. In addition, this automated system must require little to no time and capability to learn and to use. The KBIS program is designed to help alleviate this problem.

Recent efforts have concentrated on the automation of indicator assessment, particularly for space situation assessment.<sup>2</sup> These have included sophisticated data base managers as well as single domain expert systems designed to maintain indicator lists. However, as the Intelligence Community is currently structured, different centers have control over different data sources, and each exploit these data sources in slightly different ways. This necessitates the sharing of relevant information between centers and creates two important problems. It is difficult to tell exactly what information may be relevant so it is often necessary to overcompensate by sending too much information,

# KBIS ARCHITECTURE



**AUTOMATIC DATA BASE GENERATION**      **EXISTING SYSTEM**      **ACTIVE AGENTS**

FIGURE 1

and it is this overcompensation which causes such a data flow problem.

It is the hypothesis of the CAESAR extension of the KBIS program that increases in accuracy and timeliness of assessments can be made by developing a cooperative framework for a number of single domain expert systems, and this may allow for a possible reduction in data flow over communication lines. The architecture will make use of a common information exchange language that will enable the domain expert systems to request and receive information from other KBS's on a shared communication network, such as AUTODIN. As a result, this system will provide the intelligence analyst with an expanded yet focused interpretation of strategic indicators developed via cross-domain synergism, as well as relieve the data distribution and flow problems which currently exist.

As can be seen in Figure 1, the architecture has levels of components: the message processing, data base, and active agent. This is the architecture currently under implementation in the KBIS program. The data base level is separated from the other two primarily so as to allow integration of advances in the other two areas with existing data bases. This is crucial, as current knowledge based systems have been shown to be most successful when dealing with a deep, narrowly scoped domain. As such, they need to be incrementally added to an existing system in order to be operationally practical. In addition, the intelligence analysis process itself is often broken down in this same manner into subanalysis tasks so as to be handled by individual analysts, and the assessments from these subtasks are combined at a higher hierarchical level. This real world process then serves as our model for the cooperative expert system design.

The active agent concept is an important one to the architecture. Active agent components are so called as it should not matter to the other two components (message processing and data base) whether the communications from that level are from a human being or a knowledge based (expert) system. Since both theoretically will request and pass similar information, that facet should be transparent to the architecture. This concept allows for multiple types of man-machine interfaces, from existing (dumb terminal) types to future sophisticated (smart terminal, intelligent MMI) types. Also, this allows for multiple sub-domain expert systems working on the same problem in a blackboard manner, shown to be a powerful method for problem solving. The key here is that these active agents are themselves independent, and cooperation takes place only at the level of using the same communication language.<sup>3</sup> This communication language within the architecture has a highly formatted syntax, but is free as to content of fields.

The message processing level is a complex and critical aspect to this architecture. Since a large number of messages are currently passed between intelligence centers in natural language, the message processing level requires a sophisticated natural language processing capability in order to make the other two levels effective. It is primarily at this level that changes will need to be made in order to have a viable distributed network architecture. However, changes at this level are strongly linked to the methods used at the active agent level. It is the hope of this work that the internal communication language currently embodied in the Supervisor module which handles the control and information transfer between the knowledge based systems, man-machine interfaces, and DBMS can serve as the baseline for the more general case.

The general case, however, is quite broad. From raw data, to multiple levels of fusion and assessment, to hypothesis and warning generation, the types of information passed vary widely. In addition to specific information requests, broadcasting of desired goals and achieved states, and passing of reasoning chains, the general case must also include control of information transfer, including network protocols, multi-level security, communication gateways (network to network links), message routing, and routing header formats.

Since the capacity of this internal communication language is crucial to the practical success of this architecture, it bears further discussion. While numerous attempts have been made in the past to rigidly format messages so as to allow for automated data processing ease (JINTACCS, for example), these attempts always have met with limited success. Limited, in the sense that all message formats allow for at least one free text area to explain and present information which doesn't fit well in the rigid fields of the message type.

This effort makes two assumptions as to the feasibility of its internal communication language. One, that the intelligence domain has a finite number of information sources which can be categorized into types (IMINT, HUMINT, SIGINT, etc.), and these types can be rigidly mapped into a rigid syntax grammar (or vector with fixed fields). Two, these vectors can be directly converted and matched to the range of knowledge representations currently employed in knowledge based systems. Each field in a given message type is itself of fixed type but free content. Should the content of a field not match a particular bit of knowledge in a KBS, the KBS can recognize that the content is outside its current scope and either initiate some learning algorithm based upon the known type and field, or flag the input for a knowledge base maintenance function.

The scope of the CAESAR effort is the development of an experimental version of a technological capability to perform distributed time critical event assessment of foreign military activity. The experimental version will demonstrate cooperating knowledge-based systems technology for indications and warning that performs both domain and cross-domain indicator analysis through bi-directional communications. The effort will require the design and implementation of experimental software for handling communications between separate knowledge-based systems. The effort will also require the development of a structured evaluation scheme (to include test procedures, evaluation criteria, etc.) to assess the system effectiveness of the experimental version.

Much of the past research in cooperating expert systems has dealt with a tightly coupled cooperation, claiming that better performance can be achieved. While blackboard concepts are still popular, the tendency has been to sacrifice true modularity for performance in such systems.<sup>4 5</sup> In addition, much of the current literature claims that in order to be truly effective, KBSs must become much more tightly integrated with data bases, and many expert system tools are evolving in this direction.<sup>6</sup> The author supports this evolution, and agrees that such architectures enhance performance and effectiveness. However, practical considerations have forced the evolution of a loosely coupled, distributed Intelligence Community, and an

architecture which models itself on this has the greatest chance of simplified integration and success. The applicability of such an architecture, though, would seem to extend beyond the needs of the Intelligence Community to any environment where a loosely coupled architecture is advantageous, such as the space station. There is a supporting body of research supporting this concept as well.<sup>7</sup>

The KBIS program has developed a number of successful stand alone systems which tackle real world problems, most notably for space situation assessment, launch prediction, and space object identification. The success of these systems and the lessons learned as to individual architectures have provided the baseline for the CAESAR effort. As it is necessary for the persons using these systems to share information, it is the obvious next step in the evolutionary development of an overall architecture for these systems. While success seems promising based upon the current research, it cannot be overly stressed that the real world is always more complex than the most ingenuous laboratory environment. Until enough individual systems are in place in operational settings to make a test of the CAESAR architecture valid, the success of the program can only be measured against other academic and laboratory research. This evaluation will take place in the 1990 time frame.

---

1 Final Technical Report, "The Intelligent Analyst System", RADC F30602-83-C-0105, June 1985.

2 Final Technical Report, "Space Foreign Launch Assessment", RADC-TR-86-157 (S), September 1986.

3 Agha, G., ACTORS, A Model of Concurrent Computation in Distributed Systems, The MIT Press, 1986.

4 Nii, H., "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", The AI Magazine, Summer 1986.

5 Hewitt, C. and Lieberman, H., "Design Issues in Parallel Architectures for Artificial Intelligence", IEEE COMPCON 84, February 1984.

6 Howard, H., et. al., "Knowledge-Based Database Management for Expert Systems", SIGART Newsletter, No. 92, April 1985.

7 Genesereth, M., et. al, "Cooperation Without Communication", AAAI-86 Proceedings, Vol. 1, 1986.